

FIG. 1

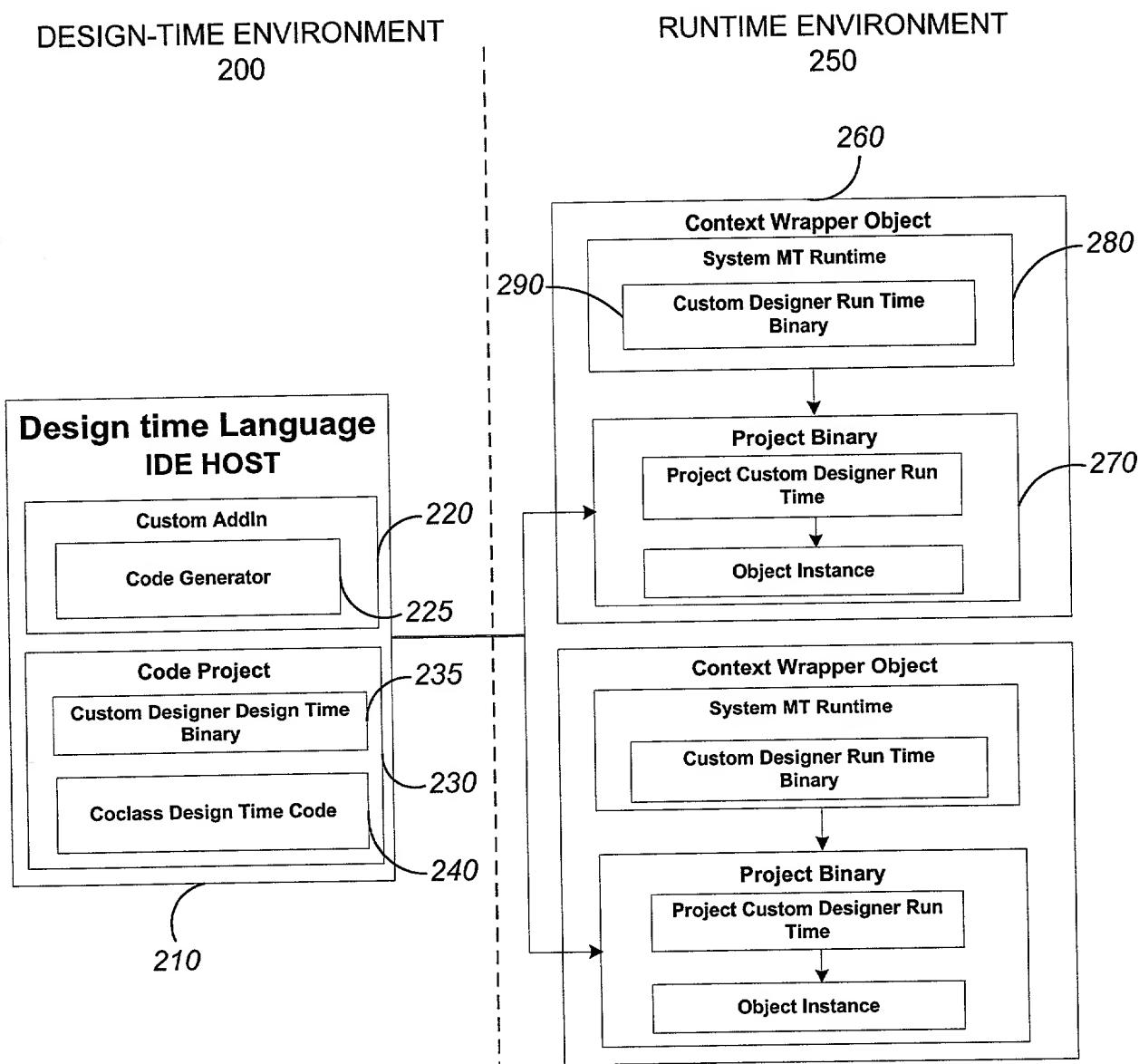


FIG. 2

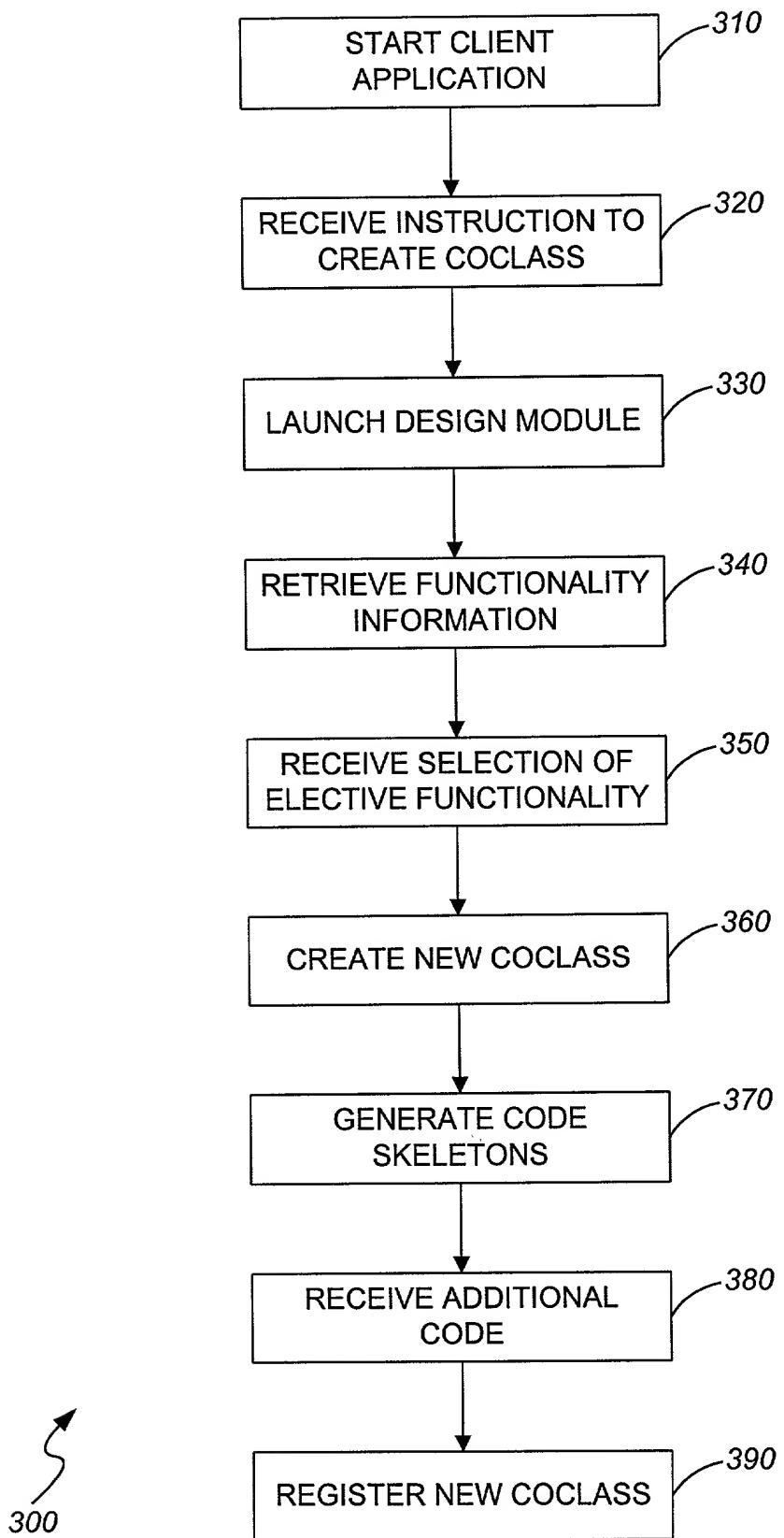


FIG. 3

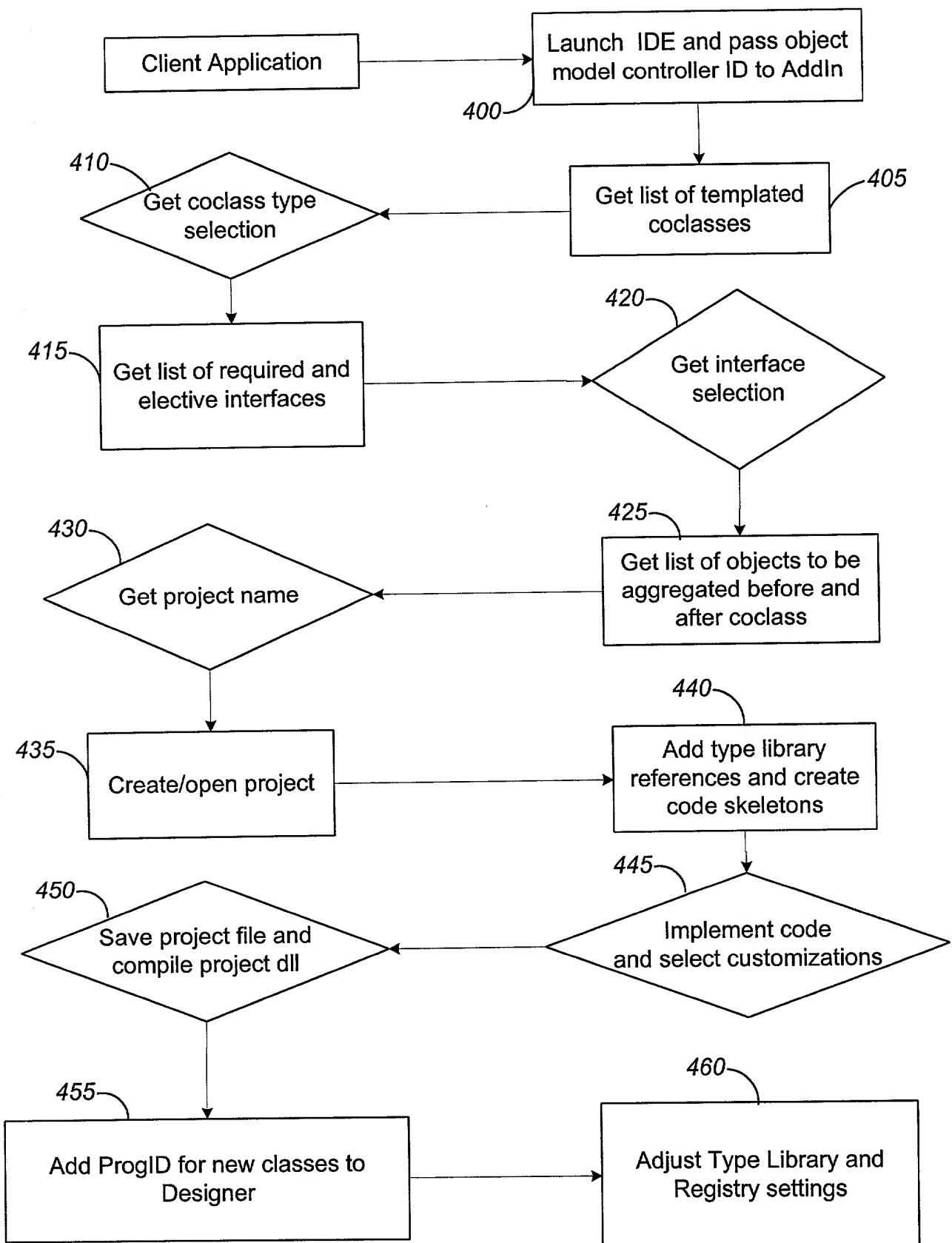


FIG. 4A

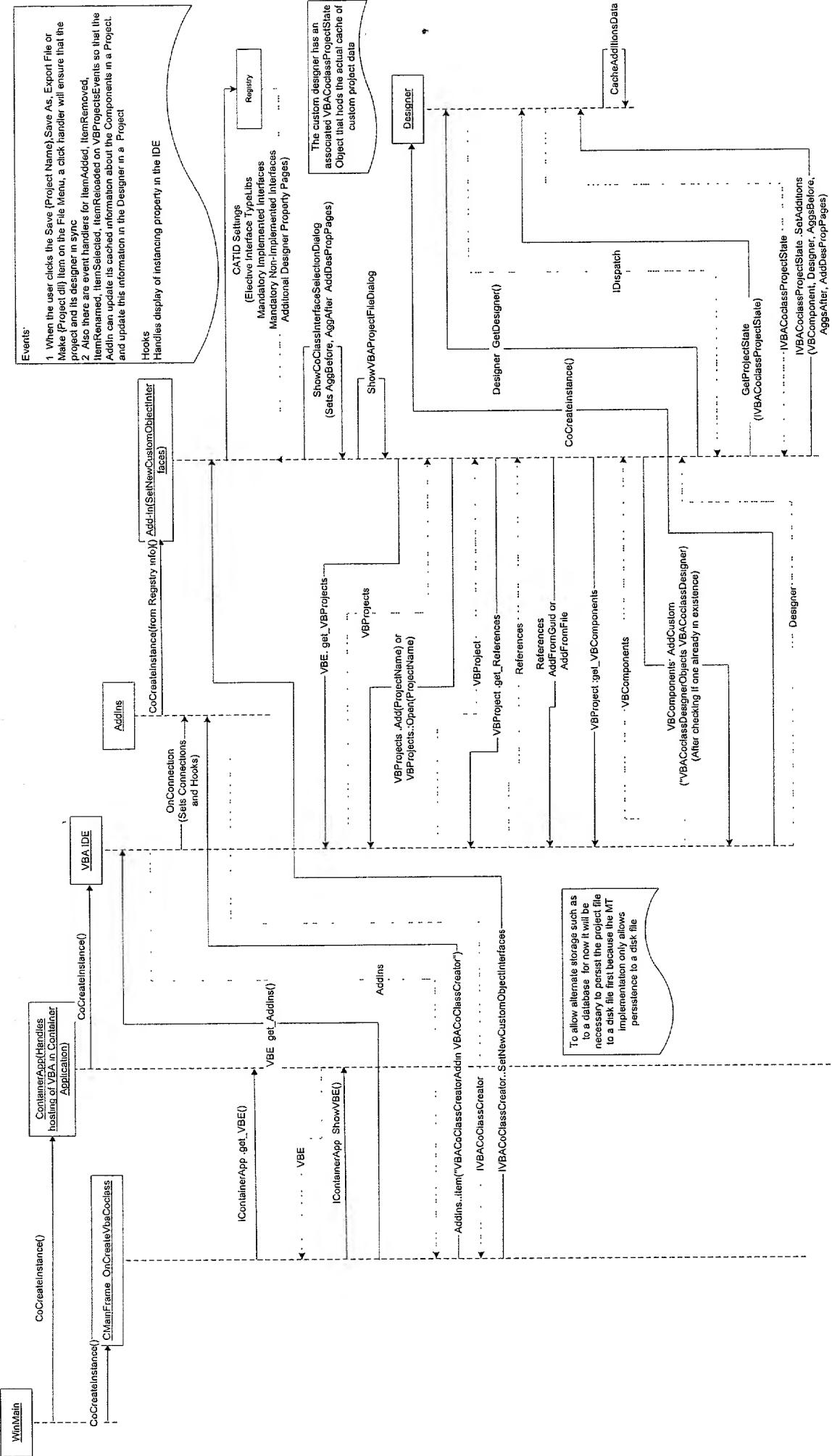


FIG. 43

VBA Project Instantiation-2

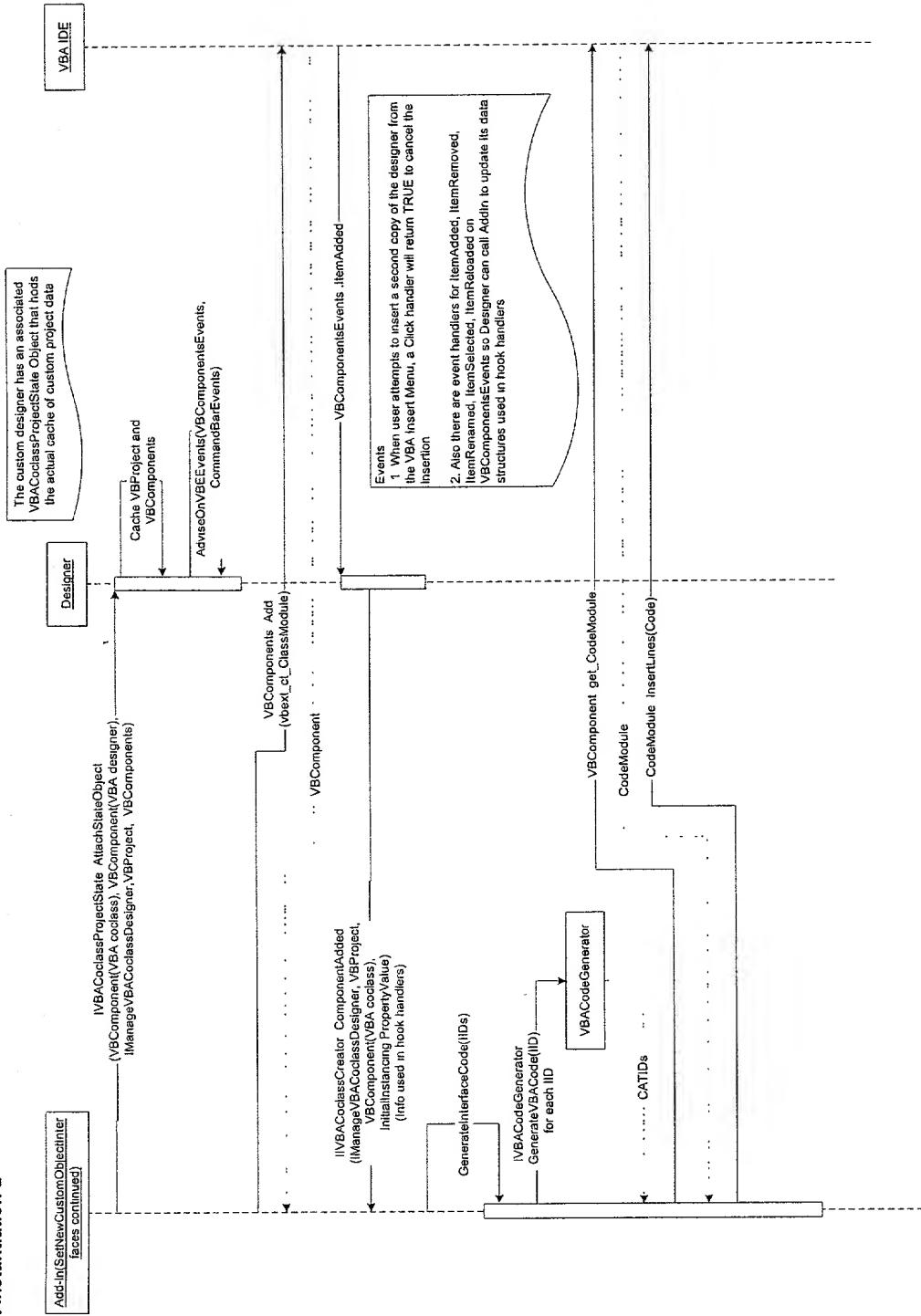


FIG. 4C

VBA Project Compilation & Save

Designer

The custom designer has an associated VBAClassPickleState Object that holds the actual cache of custom project data.

VBE:FileMenu::Save|SaveAs;
ExportFile|CloseProject;
Make

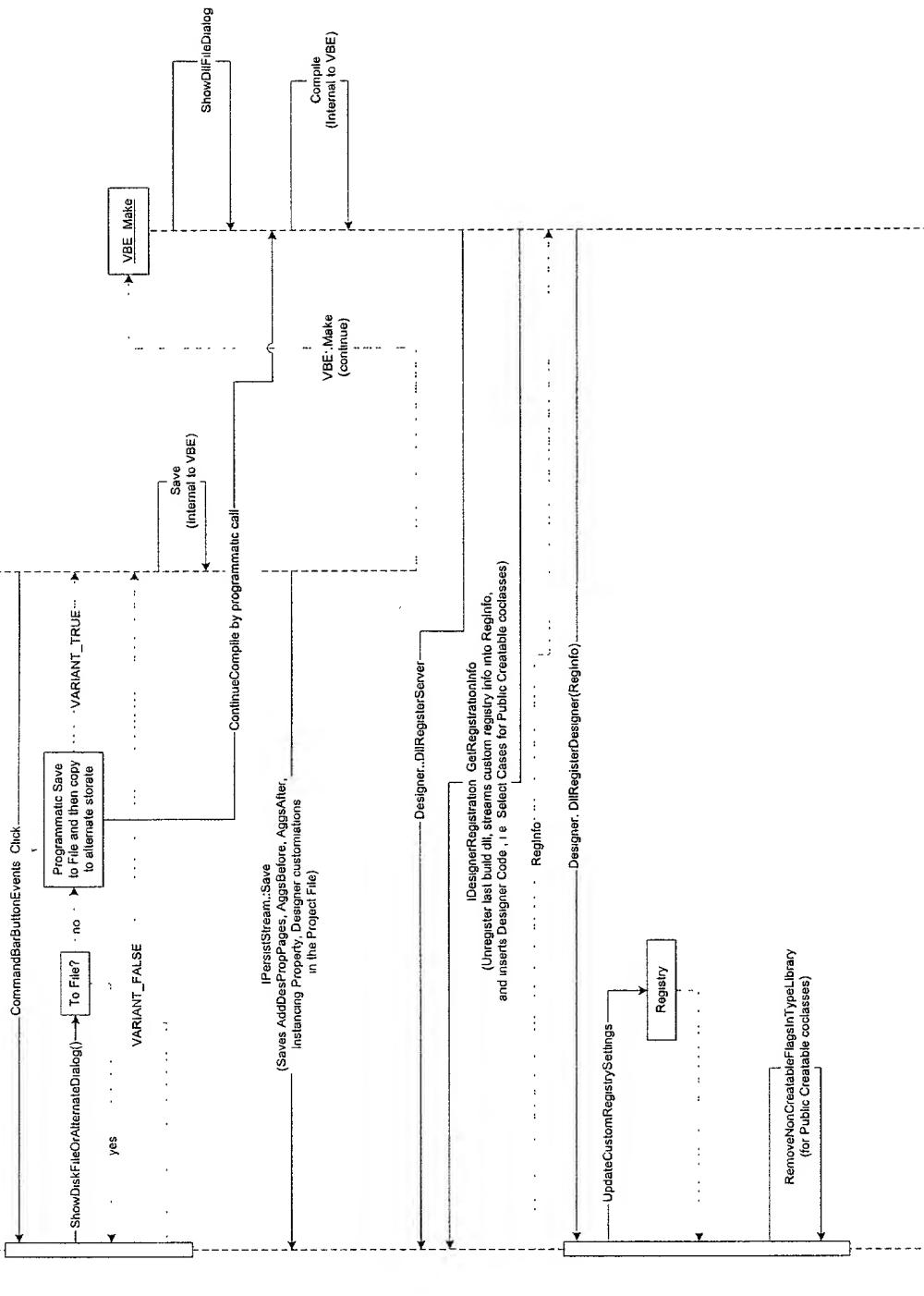


FIG. 4D

VBA Project Open, Removal

File → Open Project → ShowVBAProject → Click Open Project in File Menu

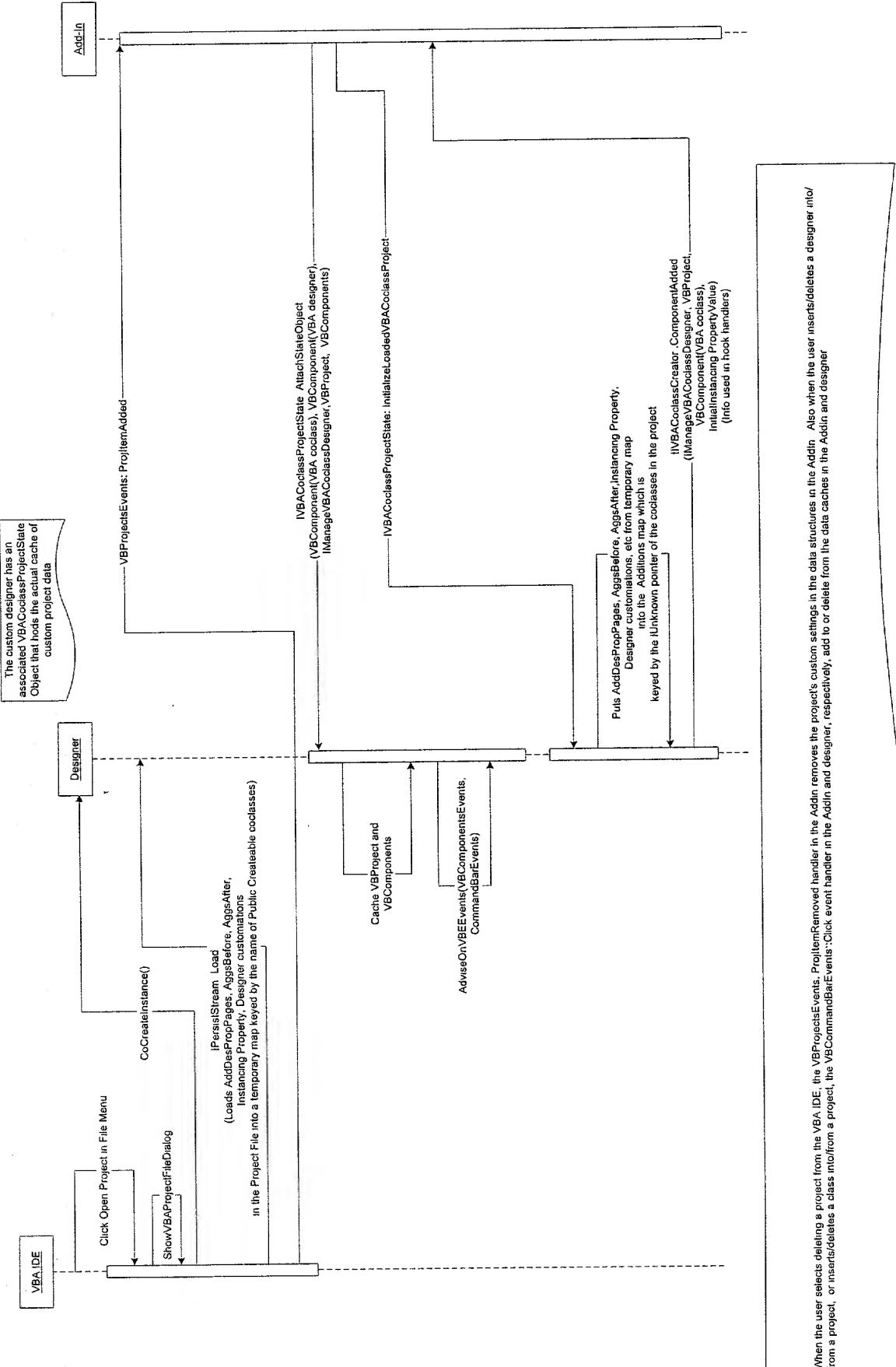


Fig. 4E

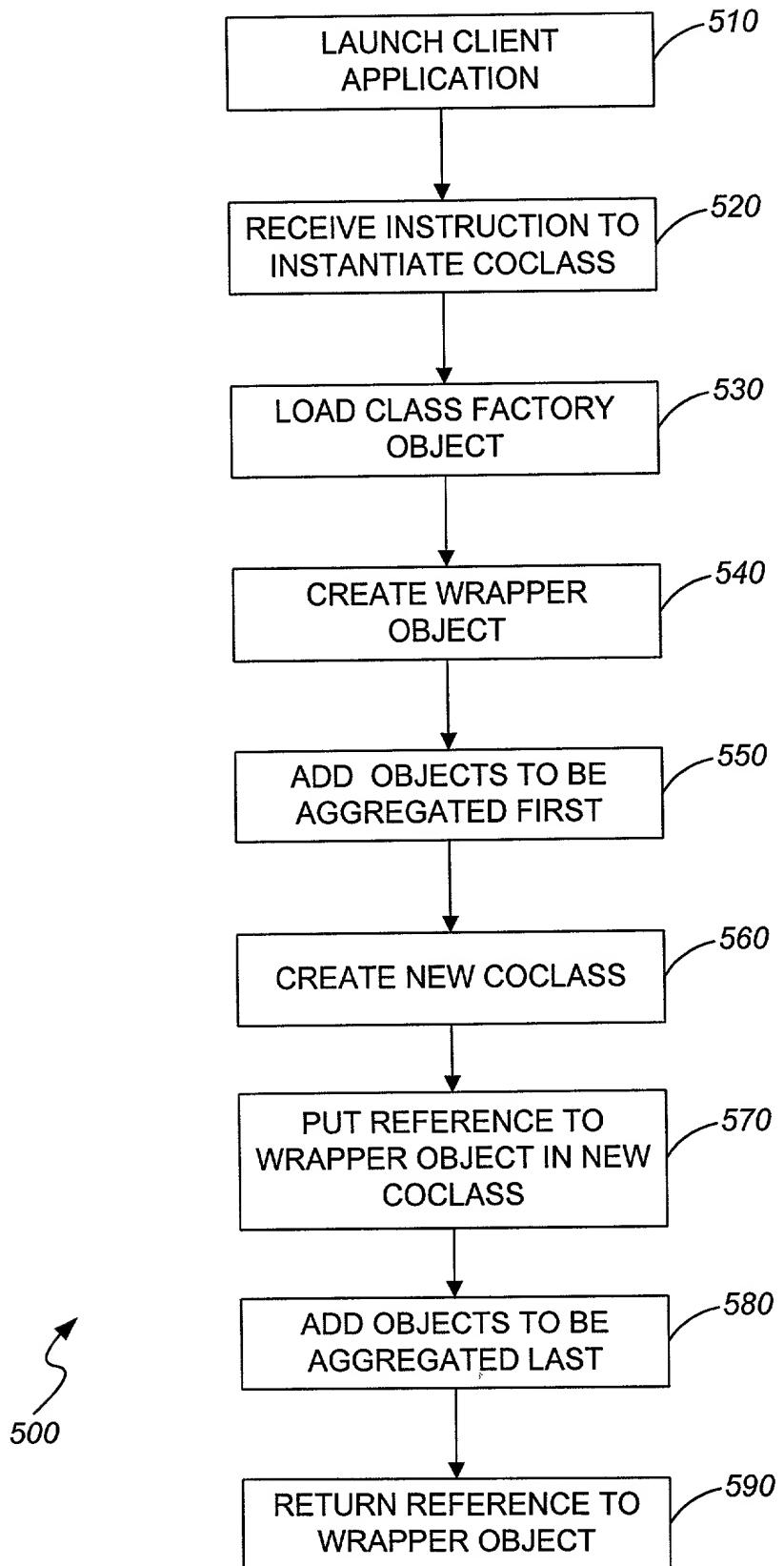


FIG. 5

Client

VBA Object Instantiation
VBACoclassWrapperObj::FinalConstruct

DllGetClassObject caches the Coclass CLSID and returns the class factory for the Wrapper object.

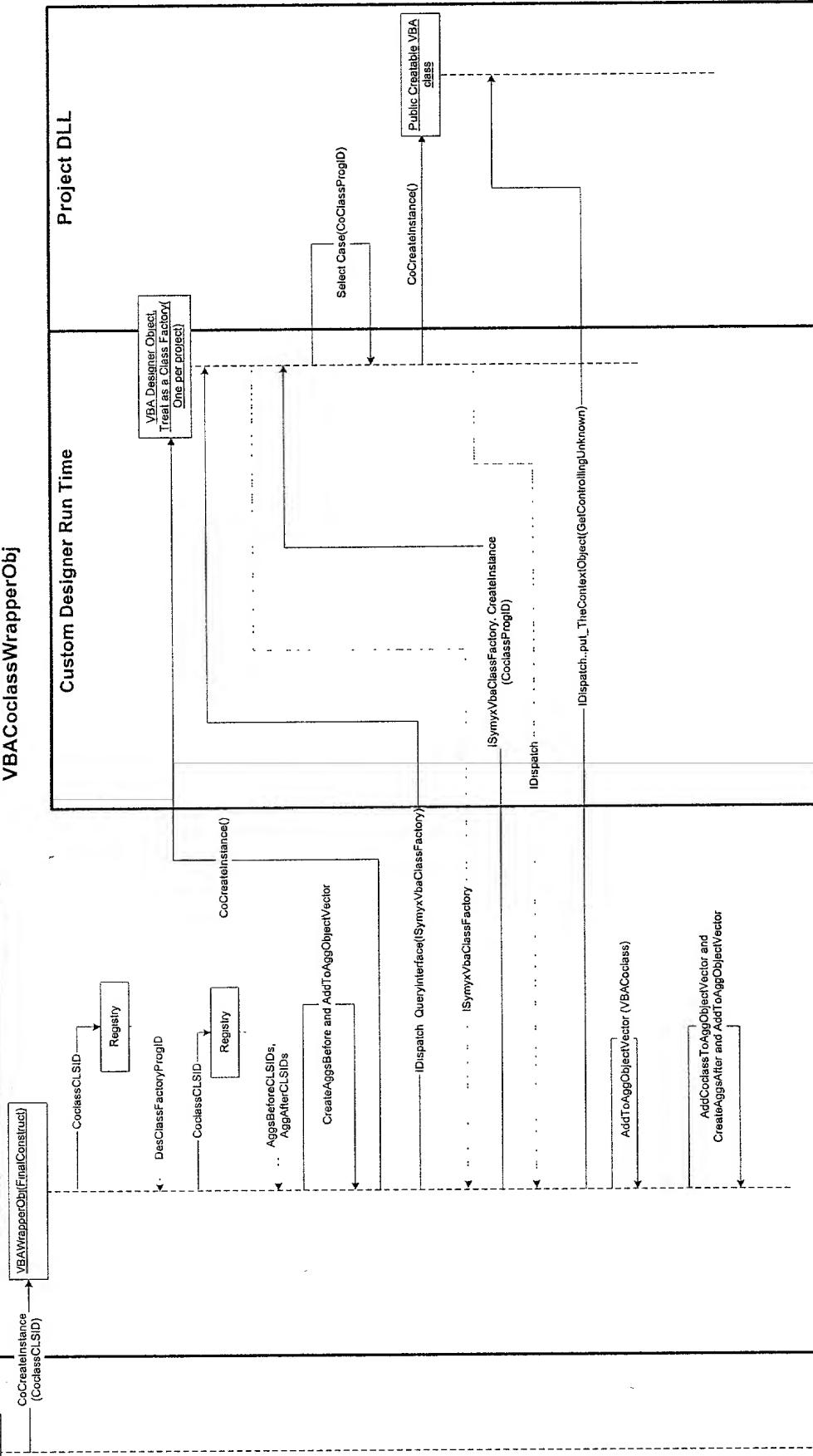


FIG. 6

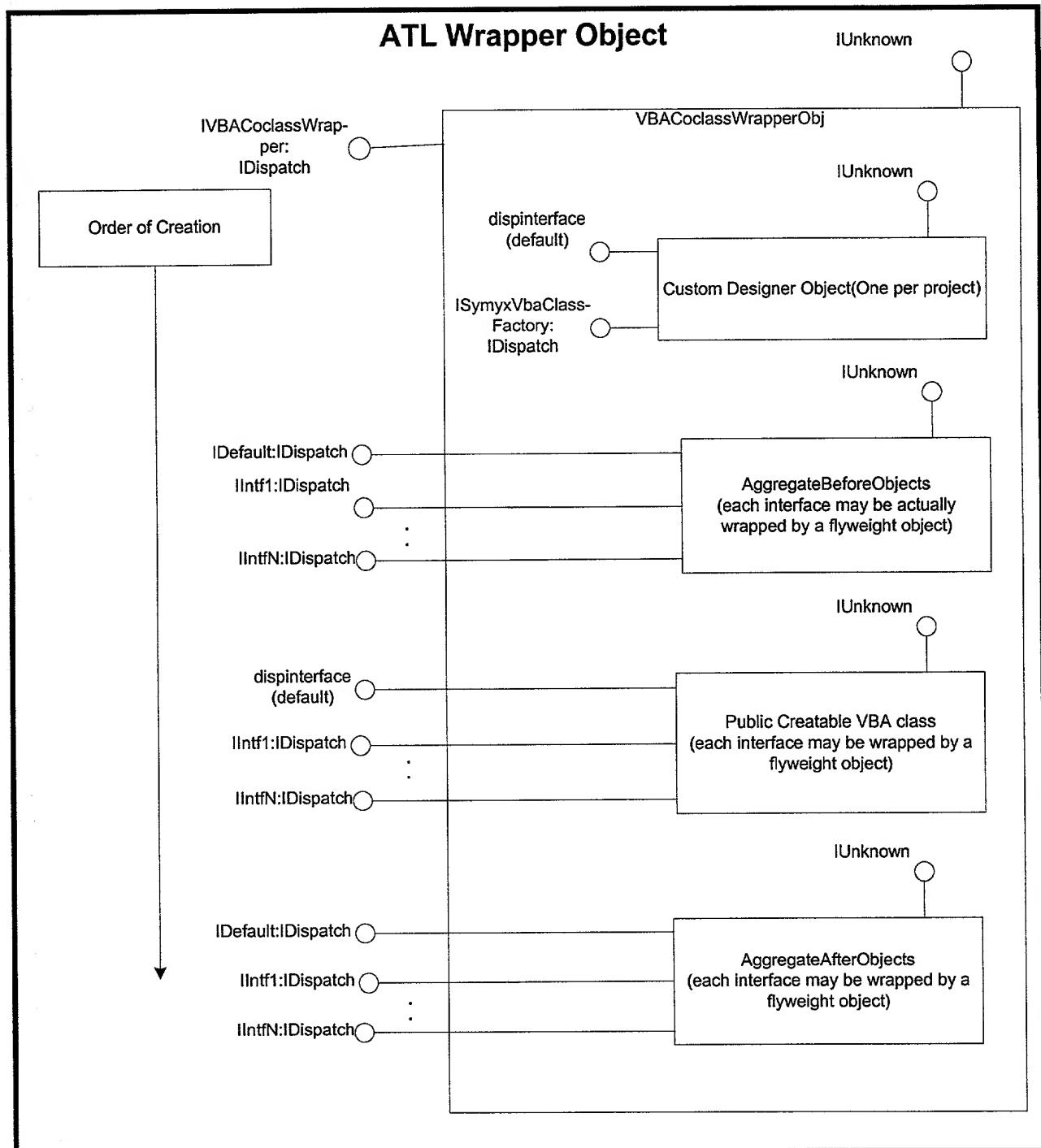
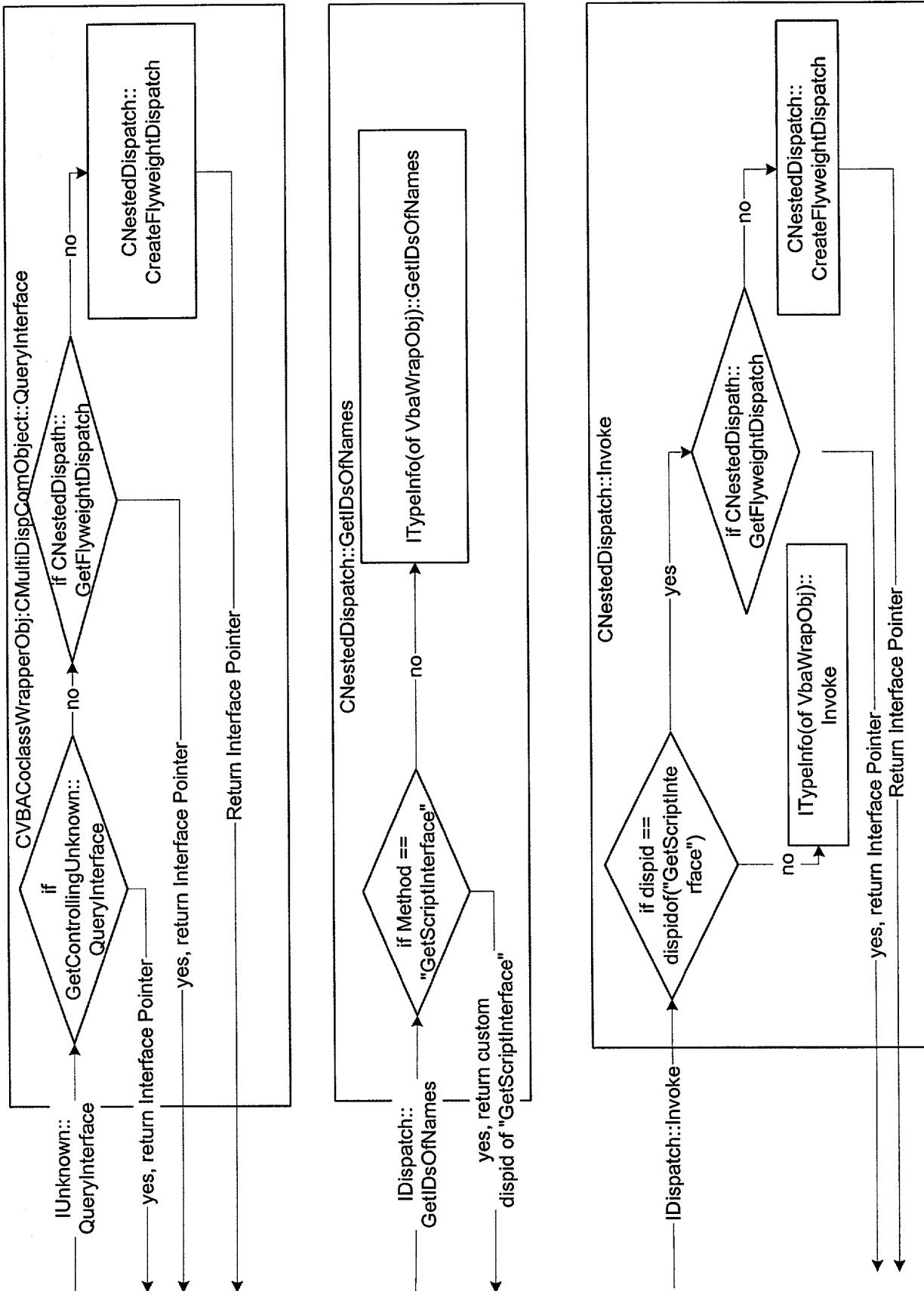
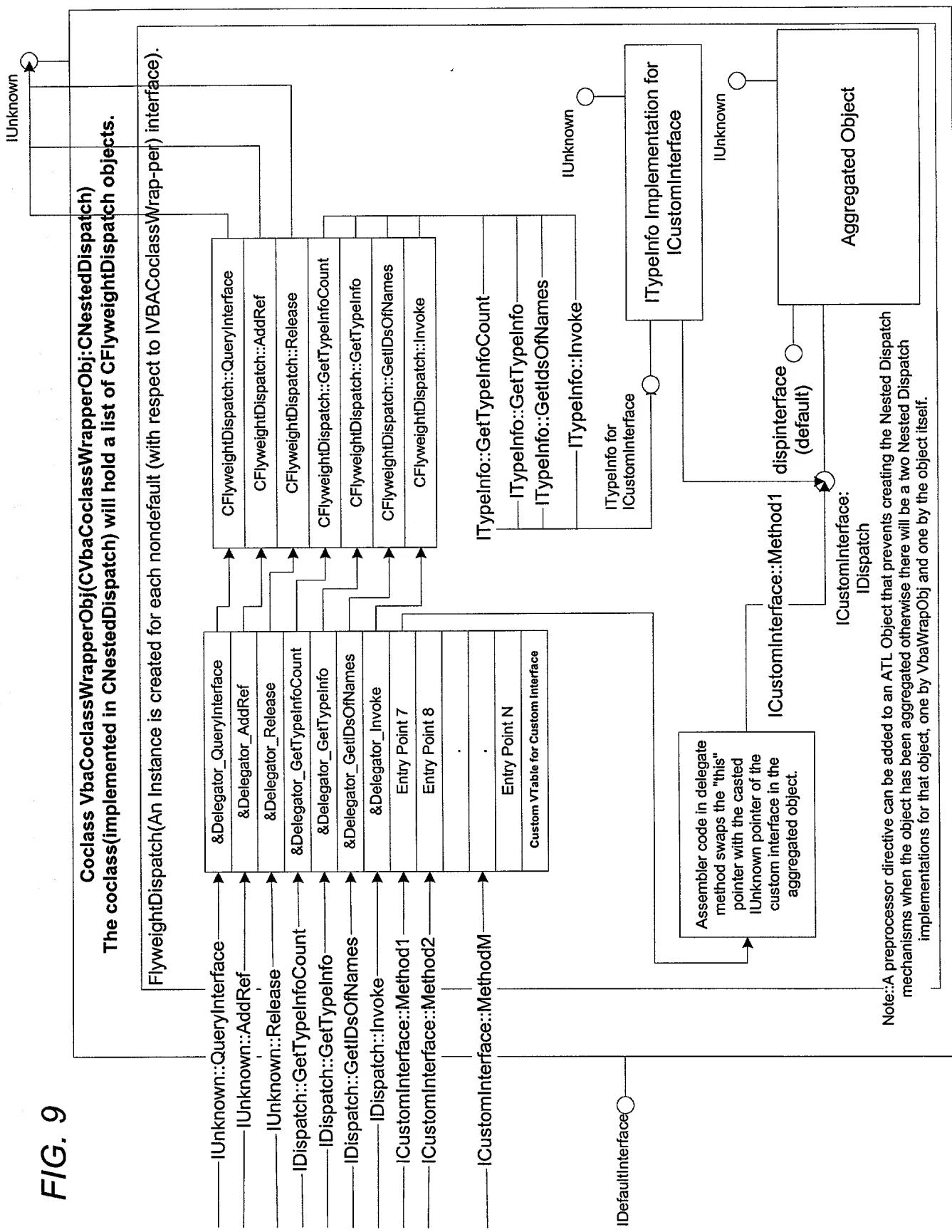


FIG. 7

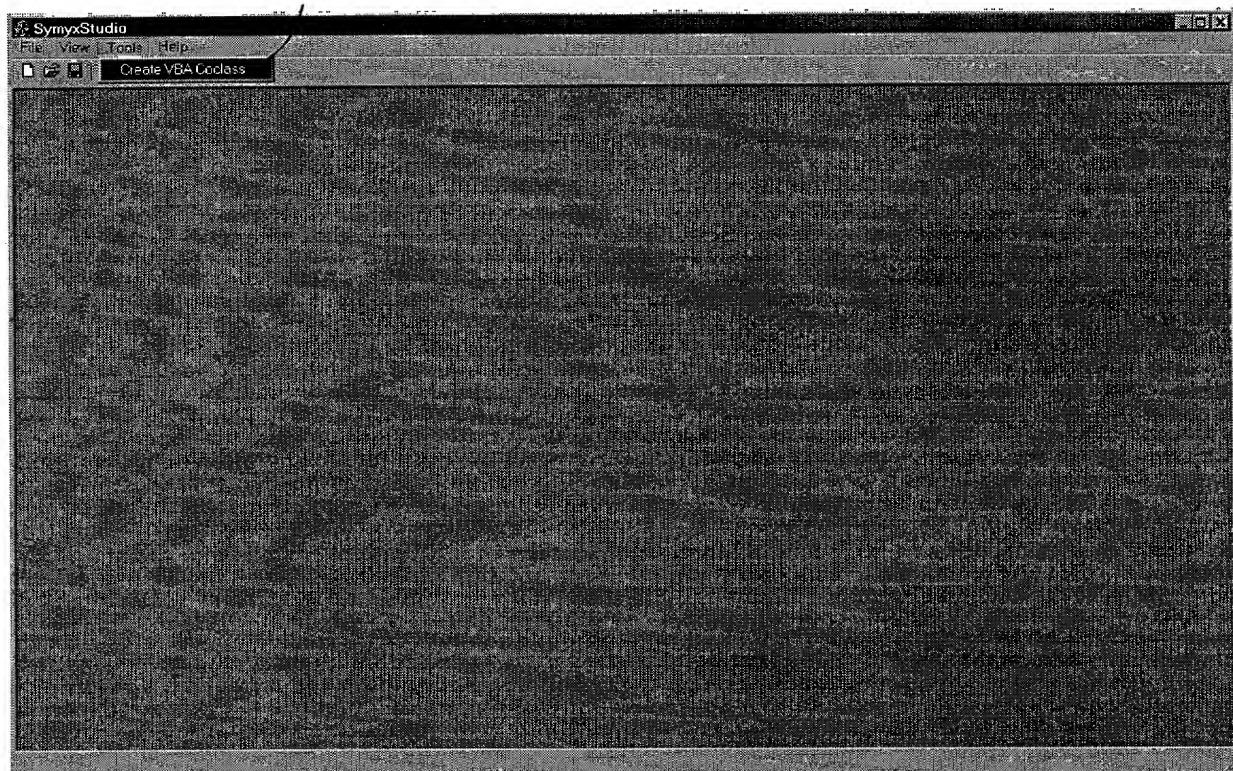


F/G. 8

F/G. 9

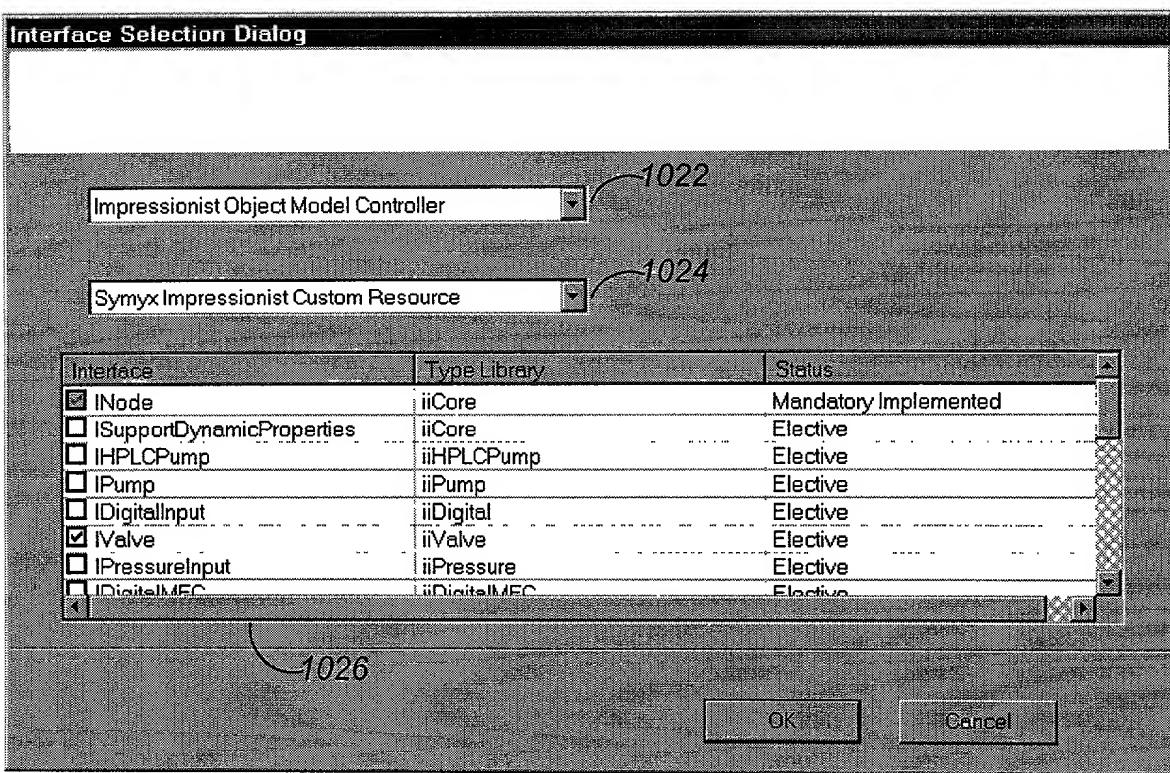


1010



1000

FIG. 10A



1020

FIG. 10B

Microsoft Visual Basic - Project1

File Edit View Insert Format Debug Run Tools Addins Window Help

Project1 - Class1 (Code)

IValve

Option Explicit

Implements IValve

Private Sub Class_Initialize()

End Sub

Private Sub Class_Terminate()

End Sub

Private Property Get IValve_Positions() As Long

End Property

Private Property Let IValve_Position(ByVal NewValue As Long)

End Property

Private Property Get IValve_Position() As Long *~1032*

End Property

Private Sub IValve_Initialize() *~1032*

End Sub

Private Sub IValve_Wait() *~1032*

End Sub

Properties - Class1

Class1 ClassModule

Alphabet Categorized

(Name) Class1

Instancing 3 - PublicCreatable

```
Code Block Content:
```

1030

FIG. 10C

00000000000000000000000000000000

The screenshot shows the Microsoft Visual Basic Editor interface. The title bar reads "Microsoft Visual Basic - F:\VBAIntegrationDemo\CustomValveProject.vba". The menu bar includes File, Edit, View, Insert, Format, Debug, Run, Tools, and Help. The toolbar has icons for New, Open, Save, Print, and Exit. The Project Explorer on the left shows a project named "CustomValveProject" with modules like "TheContextObjectModule" and "Class Modules". A class named "Class1" is selected. The code editor window displays the following VBA code:

```
Implements IValve

Private m_ValvePosition As Long
Private m_Address As IPropertyInteger
Private m_NumPositions As IPropertySingleChoice

Private Sub Class_Initialize()

    Dim AggNode As INode
    Set AggNode = TheContextObject
    AggNode.Version = 1
    AggNode.IconResID = 0
    AggNode.Name = "Dave's Custom Valve"

    Set m_Address = New PropertyInteger
    m_Address.Name = "Address"
    m_Address.XMLTag = "Address"
    m_Address.Value = 1
    AggNode.Properties.Add m_Address

    Set m_NumPositions = New PropertyEnum
    m_NumPositions.Name = "Number of Positions"
    m_NumPositions.XMLTag = "NumPositions"
    m_NumPositions.Choices = "2|4|8|16"
    m_NumPositions.Text = "2"
    AggNode.Properties.Add m_NumPositions

    m_ValvePosition = 1

End Sub

Private Property Get IValve_Positions() As Long
    IValve_Positions = CLng(m_NumPositions.Text)
End Property

Private Property Let IValve_Position(ByVal newValue As Long)
    m_ValvePosition = newValue
End Property

Private Property Get IValve_Position() As Long
    IValve_Position = m_ValvePosition
End Property
```

A handwritten note "1040" is written next to the line "Set m_Address = New PropertyInteger".

FIG. 10D

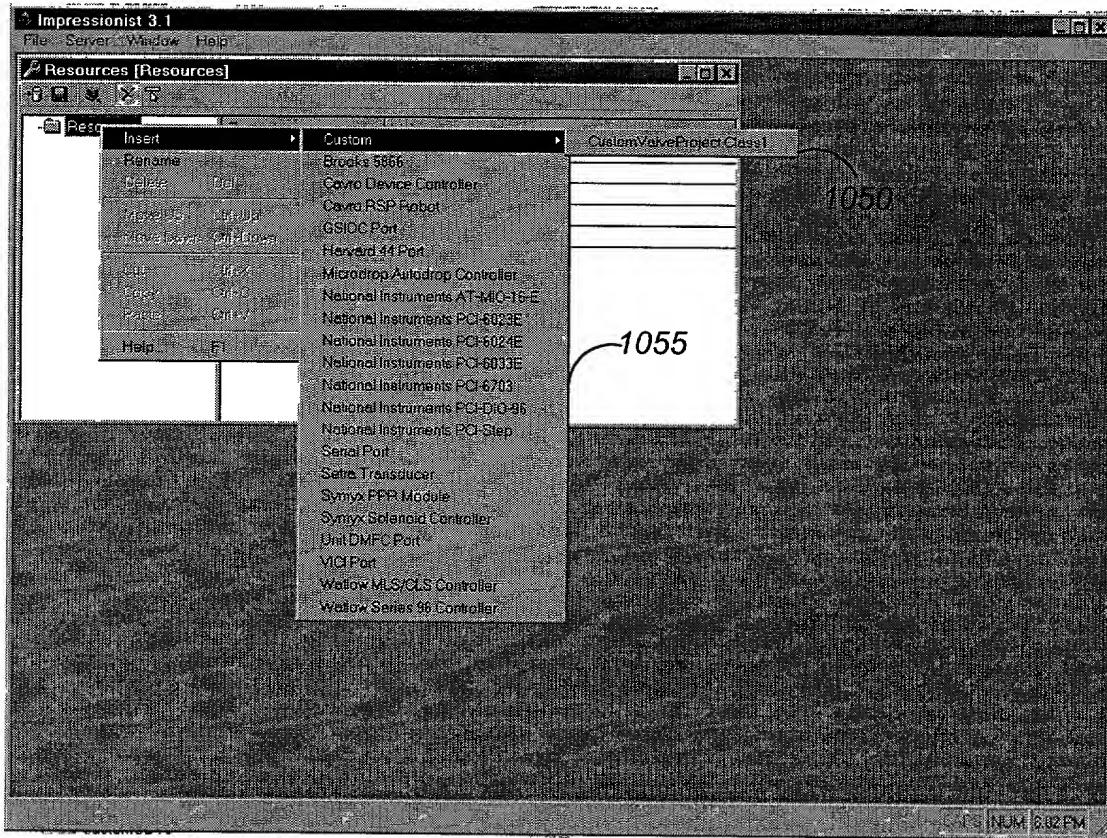


FIG. 10E

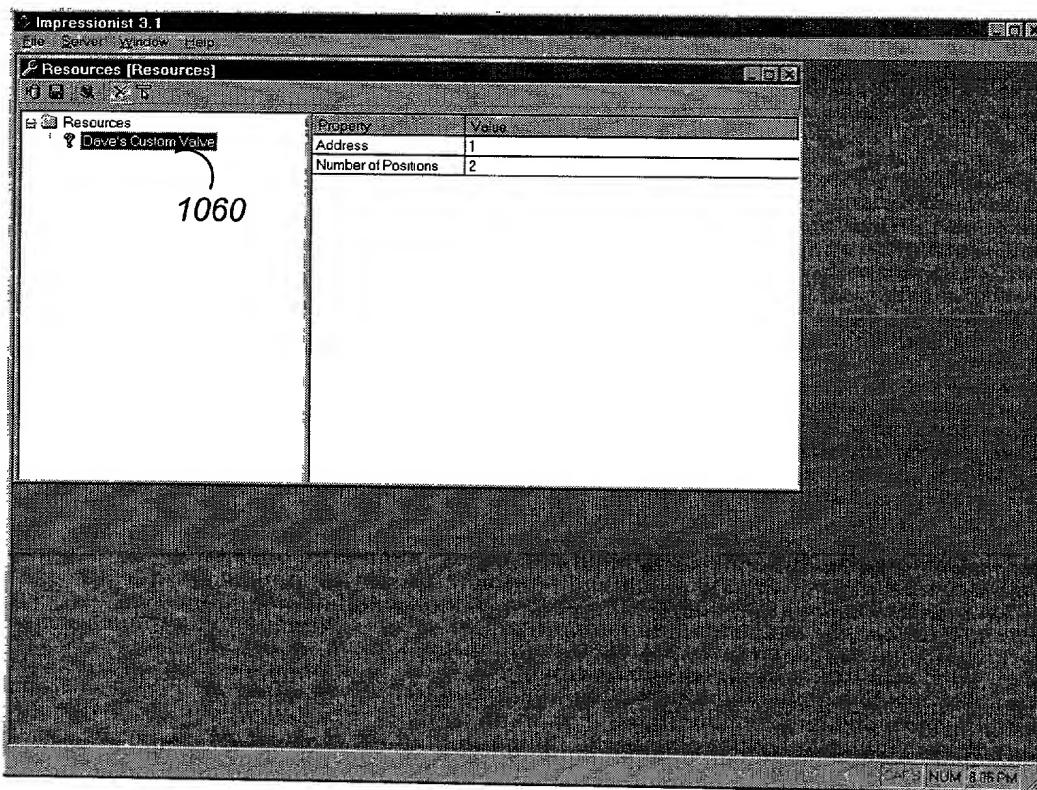


FIG. 10F